Delta Network Performance and "Hot Spot" Traffic

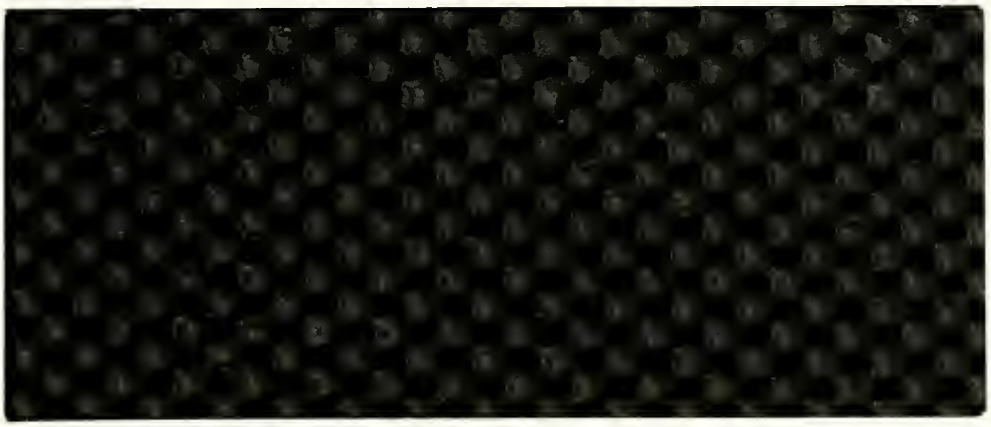
by

Yue-Sheng Liu

Ultracomputer Note #132
January, 1988

Ultracomputer Research Laboratory

New York University
Courant Institute of Mathematical Sciences
Division of Computer Science
251 Mercer Street, New York, NY 10012



Delta Network Performance and "Hot Spot" Traffic

by

Yue-Sheng Liu

Ultracomputer Note #132
January, 1988

This work was supported by National Science Foundation under grant number NSF-DCR-84-13359.

ABSTRACT

In this paper, we review previous work on both buffered and unbuffered delta network performance analysis. We then analyze the delta network performance under "hot spot" traffic and give a solution to the single "hot spot" case.

1. Introduction

Interconnection networks have been widely used in parallel processing systems. The performance of interconnection networks under uniform traffic load has been studied with both analytical methods and simulations (Patel[1981], Dias & Jump[1981], Kruskal & Snir[1983], Kumar & Jump[1984], Kruskal et al [1986]). Recently, the "hot spot" traffic problem, a non-uniform traffic load model in which one memory module is favored, has received attention (Pfister & Norton [1985]). In this paper, we give an analytical solution to network performance under "hot spot" traffic.

1.1. Delta Network

A typical shared memory multiprocessor system can be depicted (Fig.1) as N independent processor elements (PEs) connecting a set of memory modules (MMs) through an interconnection network (ICN). Examples of such systems are NYU Ultracomputer (Gottlieb et al [1983]), IBM RP3 (Pfister et al [1985]), and BBN Butterfly (Rettberg & Thomas [1986]).

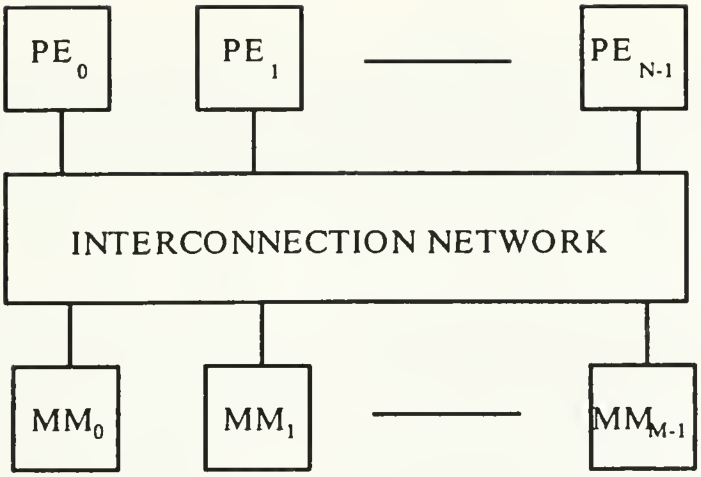
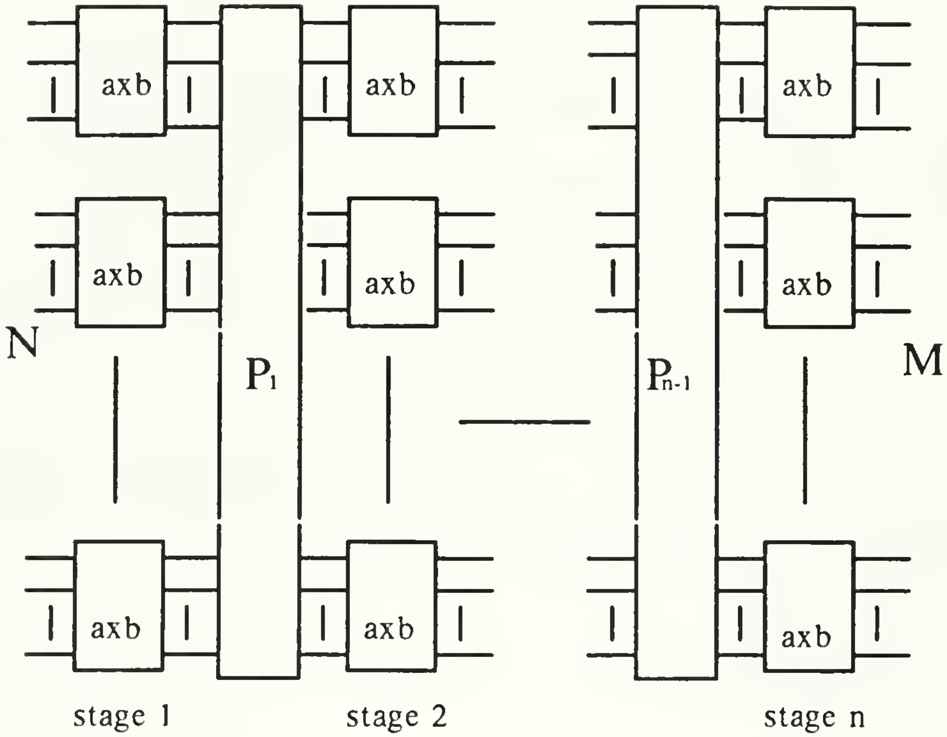


Fig.1 A Shared Memory Multiprocessor System

Interconnection networks considered here are packet switching networks consisting of multiple stages of switches. Quite a few interconnection schemes have been proposed; Feng [1981] gave a good survey. Among the various interconnection networks proposed, those most commonly used are the class of *delta networks* (Patel [1981]) which includes the omega network (Lawrie [1975]), the indirect binary n-cube network (Pease [1975]), and the cube network (Siegel & Smith [1978]).

Let an $a \times b$ switch node be an a input and b output device which has the capability to route a message from any one of the a inputs to any one of the b outputs. Let the outputs be labeled $0, 1, \ldots, b-1$. A message coming in an input terminal is routed to the output labeled d if the control digit supplied by the input message is d, where d is a base-b digit. Following Patel's definition, *a delta network* is an $a^n \times b^n$ switching network with n stages, consisting of $a \times b$ switch nodes. The link pattern between stages is such that *there exists a unique path of constant length from any source to any destination.* Moreover, the path is *digitally controlled* such that a switch node routes an input message to one of its b outputs depending on a single base-b digit from the destination address of the message. A general delta network is shown in Fig.2. An example of $4^2 \times 4^2$ delta network is depicted in Fig.3.



P_i is an $a^{n-i}b^i$ to $a^{n-i}b^i$
permutation function for $i = 1, 2, \ldots, n-1$.

Fig.2 A General Delta Network.

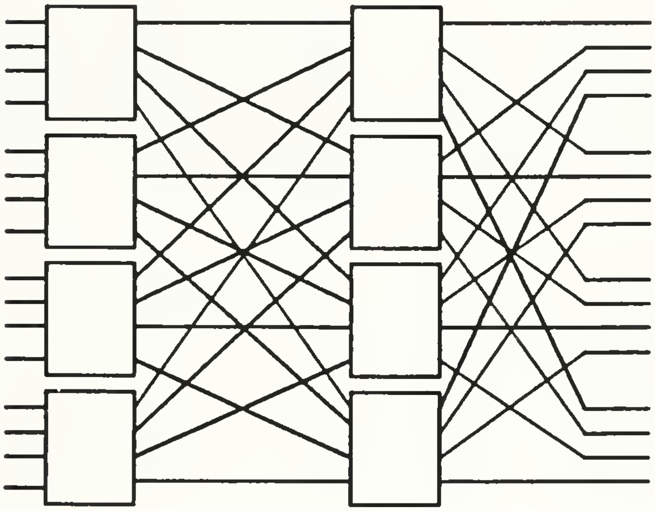


Fig.3 A $4^2 \times 4^2$ delta network

We assume that the network terminals on the PE side are sources and the other ends (to MMs) are destinations. The function of the ICN is to pass the requests from sources to the corresponding destinations according to the request addresses specified by the PEs. For simplicity, in this paper, we assume the case a=b=k, so that we have a $k^n \times k^n$ *square network*. But usually the results may be applied to general delta networks.

A network may be either *unbuffered* or *buffered*. In the *unbuffered network*, when more than one message conflict at a switch node (i.e., more than one request demand the same output port at the same time) one of them will be chosen to pass and the others rejected. In this paper we assume that the rejected messages are lost. The sender will resubmit the message when it times out. The BBN Butterfly uses such an unbuffered network. Following is an unbuffered switch:

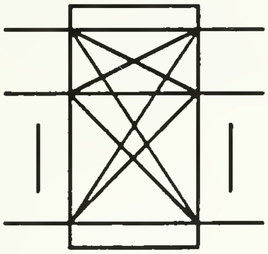


Fig.4 An Unbuffered Switch

In a *buffered network*, buffer (or queue) mechanisms are associated with input ports, output ports, or both input and output ports of a switch. Conflicts are resolved by queueing the messages. The Ultracomputer and the RP3 use buffered networks. An output buffering switch node which we will use for our analysis is depicted in the Fig.5.

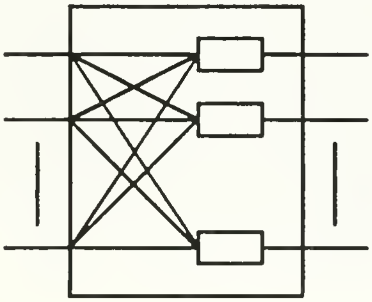


Fig.5 A Buffered Switch

1.2. Performance Analysis of Interconnection Networks

The work on analysis of interconnection network performance could be classified into two models: the *uniform traffic model* in which we assume the requests to the network are generated with their addresses uniformly distributed among all the outputs; and the *nonuniform traffic model* in which the request addresses are not uniformly distributed. As mentioned before, there are two types of switch networks: unbuffered and buffered. They also present a dimension in analyzing network performance.

1.2.1. Uniform Traffic Model

In almost all analysis of the uniform traffic model, we not only assume the uniform address distribution but also assume that the arrivals of all requests are generated as a random memoryless process. For simplicity, we only analyze one way traffic through the network. The methods are usually easy to apply to the round trip.

Unbuffered Networks

Let us first see some work on the uniform traffic model. Patel [1981] gave a recursive formula to compute the throughput of the unbuffered delta network under given offered load. This formula was also given in Dias & Jump [1981] with a simpler derivation. Assume an $N \times N$ square delta network built by $k \times k$ switches with $n = \log_k N$ stages. As shown in Fig.4, at each stage i, $i = 1, 2, \ldots, n$, assume each of k input ports of a switch has probability p_{i-1} of sending a message; and each of output ports has probability p_i of getting a message, since the requests are uniformly distributed among k output ports by the assumptions of our model. Each input port will have probability $\frac{p_{i-1}}{k}$ to send a message to an output port, and probability $(1 - \frac{p_{i-1}}{k})$ not to send. The probability that all k input ports will not send messages to a output port is $(1 - \frac{p_{i-1}}{k})^k$. This is also the probability that an output port does not get message. Therefore

$$p_i = 1 - (1 - \frac{p_{i-1}}{k})^k, \quad i = 1, 2, \ldots, n \tag{1}$$

Note that actually p_0 is the *offered load* and p_n is the *throughput*. Kruskal & Snir [1983] showed that the asymptotic behavior of the above recursive formula is

$$p_n = \frac{2k}{(k-1)n}[1 - \frac{(k+1)}{3(k-1)} \frac{\ln n}{n} + O(\frac{1}{n})] \tag{2}$$

for large enough n. Here the first and second order terms are independent of p, in which we assumed $0 < p \leq 1$. In general, this can be approximated by

$$p_n \approx \frac{2k}{(k-1)n + 2k/p}. \tag{3}$$

Buffered Networks

The analysis of buffered networks is much more complicated than the unbuffered case. In fact, the exact analysis of buffered networks with finite buffer space is still an open problem. Assume a buffered switch is as shown in Fig.5. Each queue is able to accept up to k requests in one cycle. Assuming infinite buffer size, the first stage network can be modeled as an M/D/1 queue. Based on this assumption, Kruskal & Snir [1983] obtained a formula for the average queueing delay w_1 at first stage of the network with given arrival rate p,

$$w_1 = \frac{(1-1/k)p}{2(1-p)}. \tag{4}$$

There are several approaches to calculate the delays at later stages. Kruskal & Snir [1983] approximated the input processes of later stages with memoryless processes and applied the same formula (4) to all stages of the network. Kruskal et al [1986] conjectured the later stage delays would have the form

$$w_\infty(\rho) \approx (1 + a\rho) w_1(\rho) \tag{5}$$

where ρ is the traffic intensity and a is some constant. They then use simulation to determine a and get:

$$w_\infty \approx (1 + \frac{4p}{5k}) w_1 \tag{6}$$

Christos et al [1987] proved that

$$w_i \leq w_1 + \frac{(1-p/k)^{-k} - 1}{2(1-p)}, \qquad i \geq 2. \tag{7}$$

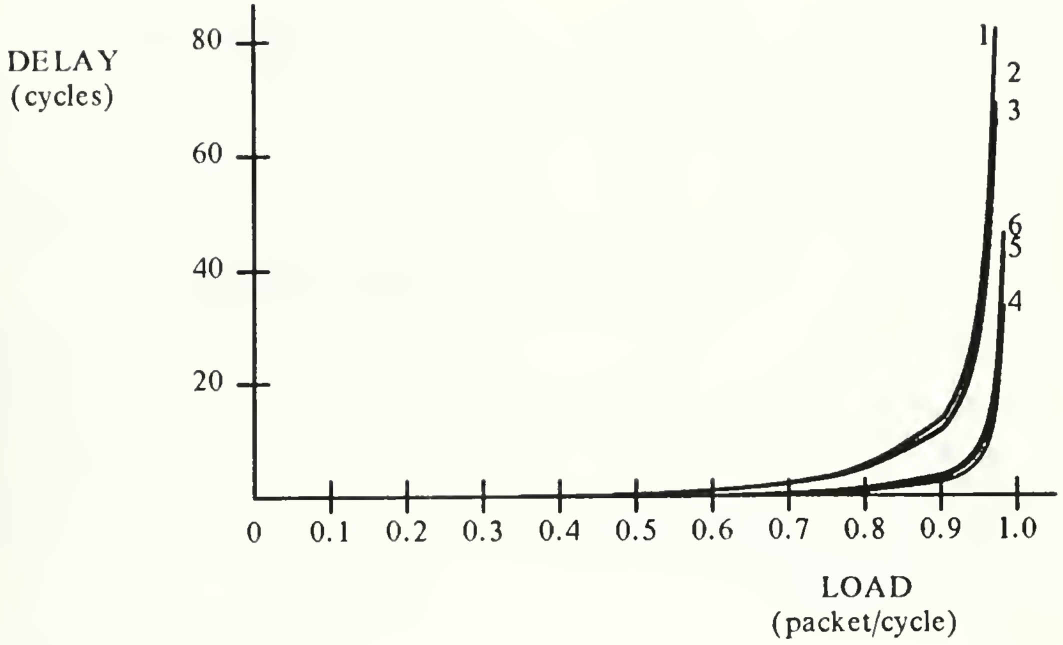


Fig.6 Results Given by Formulas (6) and (7)

We compare the results given by these two formulas in Fig.6: 1. formula (7) with k =2; 2. formula (7) with k =4; 3. formula (7) with k =8; 4. formula (6) with k =2; 5. formula (6) with k =4; and 6. formula (6) with k =8.

1.2.2. Nonuniform Traffic Models

In contrast to the uniform traffic model, in nonuniform traffic models, we assume that the request addresses are not uniformly distributed among all outputs.[1] Some outputs may be preferred over others by the requests. One particularly interesting model is the *"hot spot" traffic model* suggested by Pfister and Norton [1985].

Kumar and Pfister [1986] analyzed the onset of the "hot spot" contention. Kruskal et al [1986] analyze a single switch "hot spot" performance. In the following we will present a theorem for delta networks under the "hot spot" traffic model. From this theorem we may reduce the analysis of the "hot spot" problem to two simple cases. Then we may use the same methods as in analysis of the uniform traffic model to analyze the "hot spot" model.

---

[1] The interarrival times of the requests could be uniformly distributed. In fact, as in the uniform traffic model, we usually do assume a memoryless interarrival time distribution.

2. A General Theorem

In this section, we present a model for the delta network under single "hot spot" traffic and prove a general theorem on such a model.

Assume an $N \times N$ delta-network, built of $k \times k$ switch nodes. The network stages are numbered $1, 2, \ldots, n = \log_k N$. There are N PEs connected to the input side of the 1st stage switches, and N MMs connected to the output side of nth stage switches.

Each PE issues two types of requests: uniform requests among all N MMs, and a hot spot request to a particular MM. Assume the uniform requests are colored white, with request rate W; and the hot spot requests are colored red, with rate R. Each PE issues $p = W + R$ requests per cycle. Following the paths of the hot spot requests, we have a red traffic tree which is rooted at the hot MM, and spans N PEs as leaves.

We label each input and output of a switch with its traffic rate. The labeling scheme is as follows:

At stage 1: Initially, we label all inputs $p_{0,0}$; the outputs in the red traffic tree are labeled $p_{1,1}$; those not in the red traffic tree $p_{0,1}$.

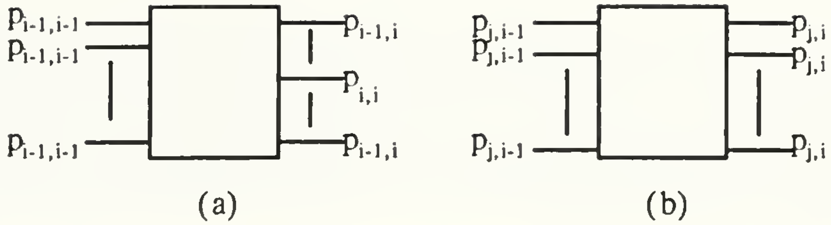
At stage $i, 1 < i \leq n$, We have have following two labeling rules:

(1) If all inputs of a switch are labeled $p_{i-1,i-1}$, then the output which leads to the hot MM is labeled $p_{i,i}$; the other k-1 outputs $p_{i-1,i}$. Such a switch will be in the hot traffic tree.

(2) If all inputs of a switch are labeled $p_{j,i-1}, j < i-1$, then all k outputs are labeled $p_{j,i}$.

Suppose we start the labeling process according to the above labeling scheme and stop only when there are no labeling rules applicable.

When the labeling process stops there are two possible outcomes: all network links are labeled; or some network links are not labeled. If the labeling process stops without all network links labeled there must be a switch node which has different input labels. Otherwise we could apply either rule (1) or (2) to proceed the labeling process. On the other hand, if all network links are labeled at the end of the labeling process, all input ports of the same switch node must have the same label. In fact, all network links can be labeled with the above labeling rules, as shown below.



(a) Switch nodes in the hot traffic tree; and
(b) Switch nodes not in the hot traffic traffic tree.

Fig.7 Two basic cases in the hot spot traffic model:

Theorem: Under the above definition, each switch node in a delta network has all its inputs labeled with the same $p_{j,i-1}$ where i is the stage of the node, and $j<i$.

Note that $i-1=j$ if and only if the node is in the red traffic tree.

Proof of the theorem:

Without loss of generality, we assume that, a message is rooted to output a, $0 \le a \le k-1$, at stage i if the i^{th} digit of the k-based destination address is a. Then we have the following Lemma.

Lemma: All traffic routed to the same switch node at stage i will have their first $i-1$ digits of the k-based destination address the same.

The lemma follows from our assumptions that there is a unique constant length path from any source to any destination and the path is specified by a k-based destination address. For any two requests meeting at stage i, if their last $n-i+1$ digits of destination address were the same they would end up at the same destination. So they must have their first $i-1$ digits of destination address the same. Otherwise we would not have a unique destination address.

Then we prove the theorem by induction:

At stage 1, it is true from the labeling definition.

Assume it it true at stage i, but not at stage $i+1$, which means that there is a switch at stage $i+1$ whose 2 inputs labeled $p_{j,i}$ and $p_{l,i}$ such that $j \neq l$.

Then from the labeling definition, we know that $j \le i$ and $l \le i$. Without loss of generality, we assume $j > l$. The input labeled $p_{j,i}$ means that this request has a destination address with j digits the same as the hot spot address. The $p_{l,i}$ request has a destination address with l digits the same as the hot spot address, but not the (l+1)st digit. Since $j \geq l+1$, $p_{l,i}$ and $p_{j,i}$ traffic flows differ at their (l+1)st digit. On the other hand, $p_{j,i}$ and $p_{l,i}$ meet at stage i+1. From the Lemma, they should have their first i digits same. This contradicts that they differ at the (l+1)st digit. QED.

The property stated by the theorem is really nice. It shows that under the single "hot spot" traffic model we assumed, the analysis of network performance could be reduced to two simple cases (corresponding to the two labeling rules): (1) those switches which are located in the hot traffic tree; and (2) those switches which are not in the hot traffic tree. For both cases all inputs of the same switch node have the same history. Therefore they have the same distribution. The switches in case (2) have the same properties as the switches in the uniform traffic model and can be analyzed with the same methods. For case (1), we have one output port preferred by all the input ports in a switch, which needs a special treatment, but no fundamental difficulty. Fig.8 is a labeled $2^3 \times 2^3$ delta network.

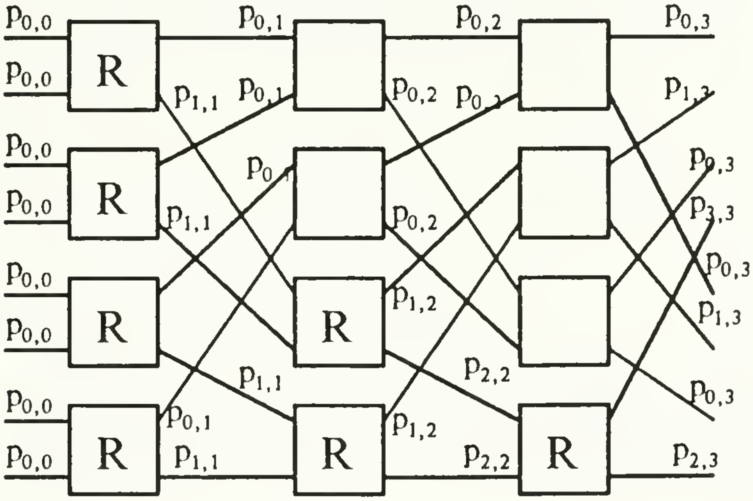


Fig.8 A labeled $2^3 \times 2^3$ delta network
with hot location at MM # 3

Furthermore, from the theorem, we may draw a picture for the network under the single "hot spot" traffic model. At the 1st stage, all the inputs are colored red and labeled $p_{0,0}$. But each switch will only have one of its outputs colored red; the total number of N/k output ports will be labeled red with $p_{1,1}$. The remaining $N(k-1)/k$ output ports will be labeled white with $p_{0,1}$. Remember that the theorem tells us that only like labeled outputs of the 1st stage will go to the inputs of the same switch at the 2nd stage. Therefore, at the 2nd stage, we have N/k^2 switches with their inputs labeled $p_{1,1}$ and colored red. The remaining $N(k-1)/k^2$ switches all have inputs labeled $p_{0,1}$. Then we apply the labeling rules to the 2nd stage, and continue, stage after stage. The result is stated in following corollary.

Corollary: Following our labeling rules above, at stage j, $1 \leq j \leq n$, the inputs of N/k^j switches are labeled $p_{j-1,j-1}$; and the inputs of $N(k-1)/k^{(i+1)}$ switches are labeled $p_{i,j-1}$, for $0 \leq i < j-1$. Then there are N/k^j outputs labeled $p_{j,j}$; and $N(k-1)/k^{(i+1)}$ outputs labeled $p_{i,j}$, for $0 \leq i < j$.

We are particularly interested in the last stage. At stage n (last stage), we have one output labeled $p_{n,n}$, and $N(k-1)/k^{(i+1)}$ outputs labeled $p_{i,n}$ for $0 \leq i < n$.

3. Unbuffered Network

Now we can apply the theorem proved above to solve the problem of unbuffered network performance under a single "hot spot" traffic model. Since there are only two basic cases, we examine each in detail. Assume that a label of a switch represents the probability that a message will be sent through the link that label is tagged. At any stage j, $1 \leq j \leq n$, all switches fall in one of two cases.

In the first case (See Fig.7a), all k inputs of a switch are labeled $p_{j-1,j-1}$. One of the k outputs is the "hot" output labeled $p_{j,j}$ and the others are labeled $p_{j-1,j}$ which are white traffic only. This is the switch node the red traffic (i.e., "hot" traffic) goes through. Assume that the red traffic ratio is r_{j-1}. Therefore, within the $p_{j-1,j-1}$ traffic presented at the switch input, there is $r_{j-1}p_{j-1,j-1}$ red traffic and $(1-r_{j-1})p_{j-1,j-1}$ white traffic. Note that the red traffic is only destinated to the "hot" output labeled $p_{j,j}$ and the white traffic is uniformly distributed to all k outputs of the switch. We first show how to compute the "hot" output $p_{j,j}$. The "hot" output consists of the red traffic part and the white traffic part. Each of k inputs has probability $r_{j-1}p_{j-1,j-1}$ of sending red traffic to the "hot" output. On the other hand the uniform white traffic of an input port has probability $(1-r_{j-1})p_{j-1,j-1}/k$ of sending a message to each of k outputs. The probability that none of k inputs send any traffic to the "hot" output is $[1-r_{j-1}p_{j-1,j-1}-(1-r_{j-1})p_{j-1,j-1}/k]^k$. Therefore we have the formula of the probability that the "hot" output has a message:

$$p_{j,j} = 1 - [(1 - r_{j-1} p_{j-1,j-1} - (1 - r_{j-1}) p_{j-1,j-1}/k]^k . \qquad (8)$$

Following a similar argument we know that the probability a white output has a message is:

$$p_{j-1,j} = 1 - [1 - (1 - r_{j-1}) p_{j-1,j-1}/k]^k \qquad (9)$$

where,

$$r_j = \frac{Rk^j}{Rk^j + W} \quad j = 0, 1, \ldots, n. \qquad (10)$$

Remember that $R + W = p_{0,0}$, where R is the request rate to the "hot spot" and W is the uniform traffic rate. With fair random resolution of collisions at outputs, equation (10) is obtained with simple calculations.

In the second case (see Fig.7b), when the switch node is not in the red traffic tree, all the inputs are labeled $p_{i,j-1}$ at stage j, for $1 \le j \le n$, $0 \le i < j-1$. By the assumption on the white traffic, all the input requests are uniformly distributed among k outputs which labeled $p_{i,j}$. Therefore we have

$$p_{i,j} = 1 - (1 - p_{i,j-1}/k)^k . \qquad (11)$$

Take a $4^5 \times 4^5$ delta network for example. In the "hot spot" traffic model, at the outputs of last stage, traffic could be divided into 6 classes:

p(5,5): "hot" traffic, one output only;

p(4,5): the traffic conflicts with the "hot" traffic for 4 stages, 3 outputs;

p(3,5): the traffic conflicts with the "hot" traffic for 3 stages, 12 outputs;

p(2,5): the traffic conflicts with the "hot" traffic for 2 stages, 48 outputs;

p(1,5): the traffic conflicts with the "hot" traffic for 1 stage, 192 outputs; and

p(0,5): the traffic has no conflict with the "hot" traffic, 768 outputs.

The effects of "hot" traffic on the other five classes of traffic are shown at Fig.8 (with $R/p_{0,0}$ = 10%) and Fig.9 (with $R/p_{0,0}$ = 5%).

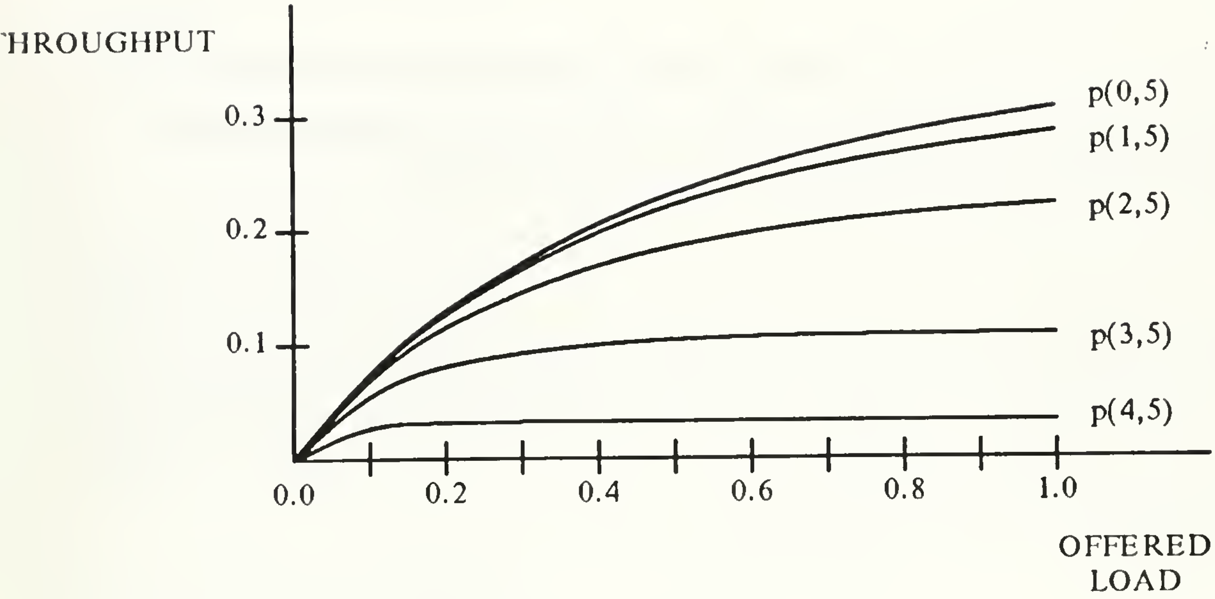


Fig.9 A $4^5 \times 4^5$ delta network performance under 10% hot spot offered load

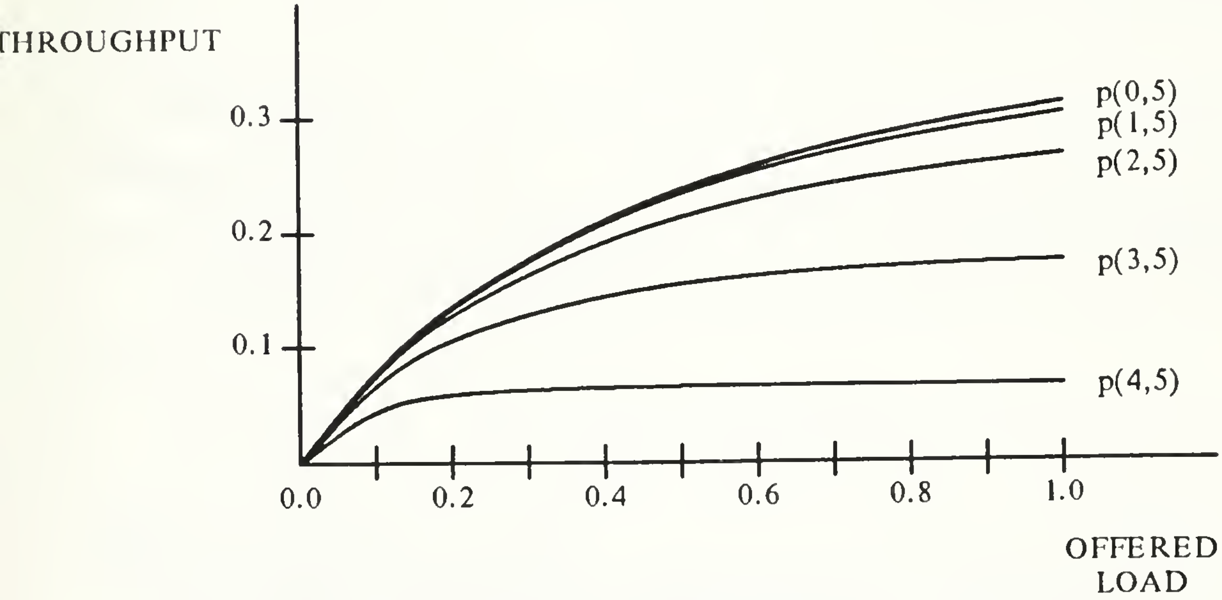


Fig.10 A $4^5 \times 4^5$ delta network performance under 5% hot spot offered load

4. Buffered Network

As we mentioned before, the problem of finite buffer space switch analysis is a hard problem, not yet solved. With the assumption of infinite buffer space we have some solutions to the problem (Kruskal & Snir [1983], Kruskal et al [1986]).

Under the "hot spot" traffic model, if we assume infinite buffer space, then the network is not *blocking*. Assume that the request rate is $p = W + R$, where W is the uniform request rate and R is the "hot spot" request rate. Since the network is not blocking, every request is able to get into the network, if the network has a stable state for a given offered load. When it reaches the stable state, at every switch node, the total input rate should be equal to the output rate. We then have

$$p_{i,i} = W + Rk^i, i = 0, 1, \ldots, n \qquad (12)$$

and

$$p_{j,i} = W, \qquad i > j, i = 1, 2, \ldots, n. \qquad (13)$$

In order for the network to be stable, we should have

$$W + NR \leq 1. \qquad (14)$$

Equations (12) and (13) are two basic conditions for the network in stable state. Applying the techniques in Kruskal et al [1986] we may obtain the delay information for the requests in the network. Due to the restriction of equation (14) we may only handle a very small amount of "hot spot" requests without combining (Gottlieb et al [1983]).

Acknowledgements

I would like to thank my advisor Allan Gottlieb for guidance. Susan Dickey has helped a lot in my work. Discussions with Richard Kenner, Wayne Berke, Pat Teller, and other people in NYU Ultracomputer Lab were helpful. Clyde Kruskal helped to derive equation (10).

References

Christos,B., Garofalakis,J., Spirakis,P., and Triantafillou,V. [1987]

"Queueing Delays in Buffered Multistage Interconnection Networks", *Proceedings of the 1987 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*

Dias,D.M. and Jump,J.R. [1981]

"Analysis and Simulation of Buffered Delta Networks", *IEEE Transactions on Computers, Vol.C-30, No.4, pp.273-282, April 1981*

Feng,Tse-yun [1981]

"A Survey of Interconnection Networks", *Computer, Dec. 1981, pp.12-27; also in Wu & Feng [1984], pp.109-124*

Goke,G.R. and Lipovski,G.J. [1973]

"Banyan Networks for Partitioning Multiprocessor Systems", *1st Annual Symposium on Computer Architecture, pp.21-28, 1973*

Gottlieb,A., Grishman,R., Kruskal,C.P., McAuliffe,K.P., Rudolph,L., and Snir,M. [1983]

"The NYU Ultracomputer -- Designing a MIMD Shared Memory Parallel Machine", *IEEE Transaction on Computers, Feb. 1983, pp.175-181*

Kruskal,C. and Snir,M. [1983]

"The Performance of Multistage Interconnection Networks for Multiprocessors", *IEEE Transactions on Computers, Vol.C-32, No.12, pp.1091-1098, December 1983*

Kruskal,C., Snir,M., and Weiss,A. [1986]

"The Distribution of Waiting Times in Clocked Multistage Interconnection Networks", *Proceedings of the 1986 International Conference on Parallel Processing*

Kumar,Manoj and Jump,J.R. [1984]

"Performance Enhancement in Buffered Delta Networks Using Crossbar Switched and Multiple Links", *Journal of Parallel and Distributed Computing, vol.1,pp.81-103,1984*

Kumar,Manoj and Pfister,Gregory [1986]

"The Onset of Hot Spot Contention", *Proceedings of the 1986 International Conference on Parallel Processing.*

Lawrie,D.H. [1975]

"Access and Alignment of Data in an Array Processor", *IEEE Transactions on Computers, Vol. c-24, No. 12, pp.1145-1155, December 1975. Also in IEEE Tutorial on Parallel Processing (ed. Kuhn,R.H. and Padua,D.A.), pp.99-109, 1981*

Pease,M.C. [1977]

"The Indirect Binary n-Cube Microprocessor Array", *IEEE Transactions on Computers, Vol.C-26, pp.458-473, May 1977*

Petel,Janak H. [1981]

"Performance of Processor-Memory Interconnections for Multiprocessors", *IEEE Transactions on Computers, C-30, Oct. 1981, pp.771-780*

Pfister,G.F. and Norton,V.A. [1985]

" "Hot Spot" Contention and Combining in Multistage Interconnection Networks", *IEEE Transactions on Computers, Vol.C-34, No.10, pp.943-948, October 1985*

Rettberg,Randall and Thomas,Robert [1986]

"Contention is no Obstacle to Shared-Memory Multiprocessing", *Communications of the ACM, vol.29,no.12, Dec. 1986*

Siegel,H.J. and Smith,S.D. [1978]

"Study of Multistage SIMD Interconnection Networks", *Proceedings 5th Annual Symposium on Computer Architecture, pp.223-229, April 1978*

Wu, Chuan-Lin and Feng,Tse-Yun (ed.) [1984]

"Tutorial: Interconnection Networks for Parallel Processing", *IEEE Computer Society Press, 1984*